

3 a.**3.a.i.**

The program aims to determine the winner of a hypothetical fight between two Marvel characters, based on which character has the higher average in six categories.

3.a.ii.

This video demonstrates the use of buttons to change screens, how the user selects characters from the drop-down box, the display changing based on the character selected, and the winner screen's outputs based on the winner.

3.a.iii.

The program's input is the different button clicks to change the screens and the character names selected from each drop-down box. The outputs of the program include the different screens, the individual rankings of the two characters, an image of both characters on the character screen, and the image of the winner (or a tie image) and that character's name (or "tie").

3 b.**3.b.i.**

```

46  onEvent (▼ "dropdown1", ▼ "change", function( ) {
47      firstCharacter = getText (▼ "dropdown1") ;
48      firstCharacterList = characterPowers (firstCharacter) ;
49      changeImageOne (firstCharacterList) ;
50      displayNumbersCharacterOne (firstCharacterList) ;
51  } ) ;

```

3.b.ii.

```

168  function findWinner (characterOne, characterTwo) {
169      var characterOneSum = 0;
170      var characterTwoSum = 0;
171      for ( var i = 0; i <= 5; i++ ) {
172          characterOneSum = characterOneSum + characterOne[i] ;
173          characterTwoSum = characterTwoSum + characterTwo[i] ;
174      }
175      var characterOneAverage = characterOneSum / 6;
176      var characterTwoAverage = characterTwoSum / 6;
177      if ( characterOneAverage > characterTwoAverage ) {
178          setImageURL (▼ "image3", characterOne[6] );
179          setText (▼ "text_areall", firstCharacter);
180      } else if ( characterTwoAverage > characterOneAverage ) {
181          setImageURL (▼ "image3", characterTwo[6] );
182          setText (▼ "text_areall", secondCharacter);
183      } else {
184          setText (▼ "text_areall", "Tie");
185          setImageURL (▼ "image3", ▼ "https://blog.knihy.cz/wp-content/uploads/2018/10/no-winner-final.jpg");
186      }
187  }

```

3.b.iii.

The name of the list being processed in this response is firstCharacterList which is used as the first parameter of the findWinner() function.

3.b.iv.

The data within the list represents the power rankings and URL for the image of the character that was selected from the first drop-down box.

3.b.v.

The list manages complexity as it allows the information for the character selected to be stored in one variable instead of seven variables which helps to condense the code needed to accomplish the goal of this program. If the list was not used more parameters would be needed for the findWinner() function and each variable would need to be added together for the first character selected before being divided by 6 to get the average. Using lists, findWinner() simply iterates through the list adding the values and displays the character image it only calls the last index of the list instead of a separate variable. Using lists allows the data to be stored, accessed, and manipulated in an easier, cleaner-looking way.

3 c.**3.c.i.**

```

168 function findWinner(characterOne, characterTwo) {
169     var characterOneSum = 0;
170     var characterTwoSum = 0;
171     for (var i = 0; i <= 5; i++) {
172         characterOneSum = characterOneSum + characterOne[i];
173         characterTwoSum = characterTwoSum + characterTwo[i];
174     }
175     var characterOneAverage = characterOneSum/6;
176     var characterTwoAverage = characterTwoSum/6;
177     if (characterOneAverage > characterTwoAverage) {
178         setImageURL(▼"image3", characterOne[6]);
179         setText(▼"text_areall", firstCharacter);
180     } else if (characterTwoAverage > characterOneAverage) {
181         setImageURL(▼"image3", characterTwo[6]);
182         setText(▼"text_areall", secondCharacter);
183     } else {
184         setText(▼"text_areall", "Tie");
185         setImageURL(▼"image3", ▼"https://blog.knihy.cz/wp-content/uploads/2018/10/no-winner-final.jpg");
186     }
187 }

```

3.c.ii.

```

65 //Makes the button that says "Click to see Winner!" on the second (character)
66 //screen to change to the third (winner) screen and used the previously stored character
67 //lists to call the findWinner function
68 onEvent(▼"Winner_Button", ▼"click", function() {
69     findWinner(firstCharacterList, secondCharacterList);
70     setScreen(▼"Winner_Screen");
71 });

```

3.c.iii.

The function takes in the character lists and compares their average power rankings. Changing the winner screen to have an image and name of the character with the higher average but if the averages are the same a separate image and the word "tie" will be displayed instead. This adds to the functionality of the program as it determines the hypothetical winner in a battle between two Marvel Characters.

3.c.iv.

The function takes in two parameters, one list for each character selected from the drop-down box. It then declares two different variables to zero representing the sums of character rankings stored in each individual list passed through. The sum for each character is determined by iterating through the first 6 elements of the list using a for loop adding each index value to the sum variable for each character. Both of these sums are then divided by 6 to get the average ranking for each. Utilizing a conditional if-else statement the averages are compared to determine the winner. If the character selected from the first drop-down box has the higher average, the name selected from that drop-down is displayed on the winner screen along with that character's image from the URL stored in the last index of the characters list. If the character selected from the second drop-down box has a higher average, that character's name and image are displayed. If the character averages are the same the words "Tie" are displayed on the screen along with an image to represent no winner.

3 d.

3.d.i.

First call:

The first call to the program has Vision selected from the first drop-down box and Bishop is selected from the second drop-down box.

Second call:

The second call to the program has the character Carnage selected from the first drop-down box and Venom from the second drop-down box.

3 d.ii.

Condition(s) tested by first call:

The first call tests the first part of the if-else statement within the findWinner() function, making sure that when the first character has a higher average than the character's name and the image is displayed on the winner screen. In this case, the winner screen should display the image of the character Vision and the text "Vision" at the bottom.

Condition(s) tested by second call:

The second call tests the third part of the if-else statement within the findWinner() function, making sure that when the averages are equal for both characters the code runs, changing the winner screen to have an image that represents no winner with the words "Tie" at the bottom.

3.d.iii.

Results of the first call:

The winner screen displays the image of Vision and the text "Vision" at the bottom.

Results of the second call:

The winner screen displays the tie image and the text "Tie" at the bottom.